

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

(NASA-CR-168939) AUTONOMOUS SCHEDULING
TECHNOLOGY FOR EARTH ORBITAL MISSIONS (Bowie
Stat Coll., Md.) 67 p HC A04/MF A01

N82-29217

CSCL 05A

GJ/81 Unclass
26009

REPORT

TO

THE NATIONAL AERONAUTICS AND SPACE AGENCY

FOR

"AUTONOMOUS SCHEDULING TECHNOLOGY FOR EARTH ORBITAL MISSIONS"

DR. SADANANAD SRTVASTAVA

BOWIE STATE COLLEGE

**ORIGINAL PAGE IS
OF POOR QUALITY**

Preface

This document is a report of the results of an initial study to develop a Dynamic Autonomous System Scheduler (DYASS) for the NASA Space Data System. Applications of artificial intelligence concepts to scheduling problems, which heretofore have been approached almost exclusively with operations research techniques, are discussed. This work was performed under contract number NGR21-027-004.

G Wiseman/100.2



National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland
20771

Official Business

Penalty for Private Use, \$300

*Maurry
6-83
5-149*



I. Introduction

The purpose of this study is to identify and discuss considerations necessary for the development of a Dynamic Autonomous System (DYASS) of resources for the mission support of near-Earth NASA spacecraft by Goddard Space Flight Center.

This report consists of six sections. The second section describes the current NASA Space Data System from a functional perspective. The future (late 80's and early 90's) NASA Space Data System is discussed briefly in the third section. The DYASS concept, the Autonomous Process Control and the NASA Space Data System are introduced in the fourth section. The fifth section surveys scheduling and related disciplines. DYASS as a scheduling problem is also discussed.

The sixth section deals with Artificial Intelligence and Knowledge Representation. Finally, the "NUDGE" System and the I-Space System are visited.

II. The Current NASA Space Data System

The NASA Space Data System can be logically partitioned into the following five major system functional elements:

- A. The constellation of spacecraft
- B. The communications and data acquisition network
- C. Mission planning and scheduling
- D. Ground processing
- E. End Users.

Figure 1 illustrates the conceptual structure and data flow of the system.

A. Constellation of Spacecraft

The Constellation of Spacecraft element performs the following functions:

1. Support of spacecraft sensor data collection
2. Provision of stored command control for sensors and spacecraft
3. Provision of data for orbit determination
4. Provision of on-board telemetry processing
5. Provision of attitude data.

The sensors represent the primary justification for the existence of the spacecraft. Telemetry data are relayed from the satellites to the ground data acquisition stations. Commands for the control and operation of the spacecraft are uplinked and executed or stored in a command memory for later execution. Tracking data such as that provided by a Range and Range Rate system communicating with the spacecraft or the LASER Network or optical sightings are used for later determination of its position in space.

For the purpose of this study, the spacecraft can be assigned to six broad categories which are useful for describing the complexity of the

spacecraft or some other special characteristic. They are:

- a. Explorers
- b. Observatories
- c. Technology development
- d. Earth resources and meteorological
- e. Foreign
- f. Other Government agencies.

The spacecraft within each of these categories are provided support from either shared or dedicated facilities. Project Operations Control Centers (POCC's) and Sensor Data Processing Facilities (SDPF) are shared by most Explorers while dedicated facilities are used by the Project for the Observatories and the Earth Resources and Meteorological satellites. The Data Acquisition and Communications support is shared by all spacecraft and is provided by the STDN and the NASCOM network, respectively. Likewise, routine orbit and attitude determination are carried out in shared facilities.

An evaluation of the list of 34 supported spacecraft now in orbit reveals that 11 are Explorers, 14 are Observatories, 4 are Technology development, 8 are Earth resources and meteorological, 3 are Foreign, and 7 are Other Government agencies.

B. Communication and Data Acquisition Network

The Communication and Data Acquisition Network includes the communications hardware and is responsible for the transfer of data between spacecraft and ground processing facilities. The ground stations acquire the telemetry data and uplink commands by the use of antennas and antenna-tracking equipment which maintain contact with the spacecraft.

ORIGINAL PAGE IS
OF POOR QUALITY

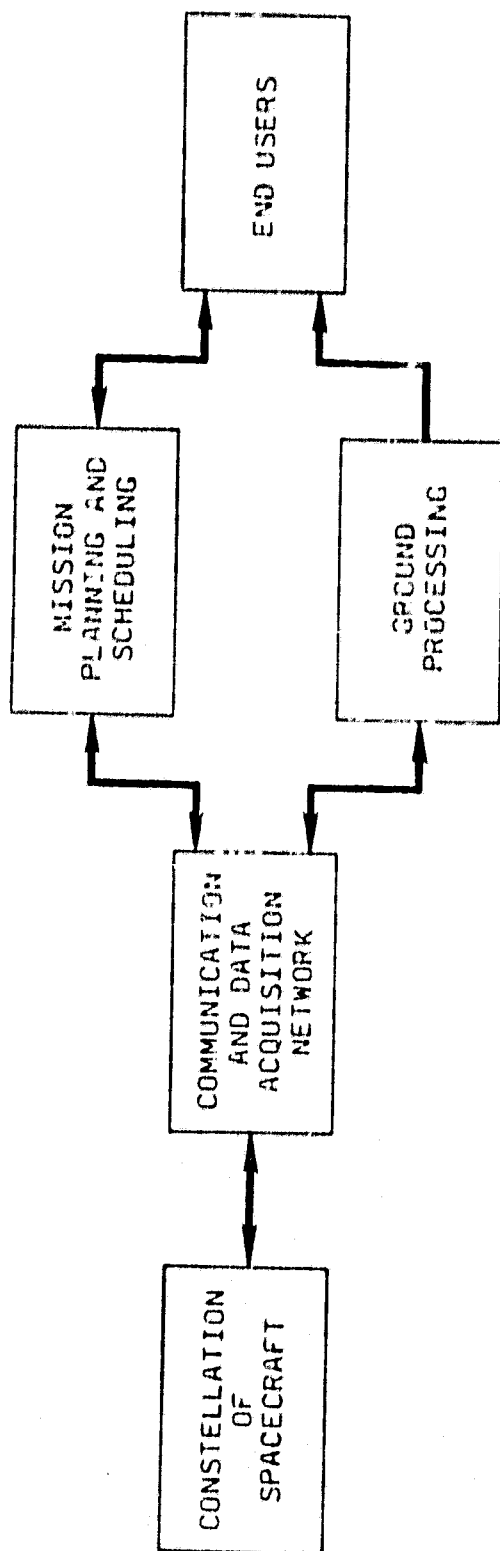


FIG. 1 NASA SPACE DATA SYSTEM MAJOR ELEMENT STRUCTURE

The NASCOM communication network allows data to be transferred between ground stations and ground processing facilities and thus makes spacecraft-to-ground communication and control possible. All of the equipment involved in this network must be scheduled to ensure proper coordination and use.

Structure

Figure 2 illustrates the structure of the Communications and Data Acquisition Network. The two functional components, the STDN and the NASCOM network, are discussed in the succeeding paragraphs.

The STDN consists of 14 ground stations which receive telemetry from and transmit commands to the spacecraft, and provide tracking data for orbit determination. Normal service at a ground station begins with a pre-pass setup period to integrate software and communications systems. During this time, the ground station hardware and the linkage between the ground station and the POCC are checked, and antenna positioning occurs. Spacecraft contact refers to the period between Acquisition of Signal (AOS) and Loss of Signal (LOS). AOS occurs at the point above the horizon where communication between the spacecraft and the ground station occurs. LOS occurs at or before the horizon, at which time communication between the spacecraft and ground station is terminated or lost.

The NASCOM network is the set of circuits, voice, switching and terminal facilities arranged in a global communication system. The system supports routine low speed and teletype data and high speed data transfer; wide band data communications channels are used for data transfer. Communications satellites are used to provide some of the high speed data transfer capability. The capacity is usually sufficient to meet the current needs of the system.

ORIGINAL PAGE IS
OF POOR QUALITY

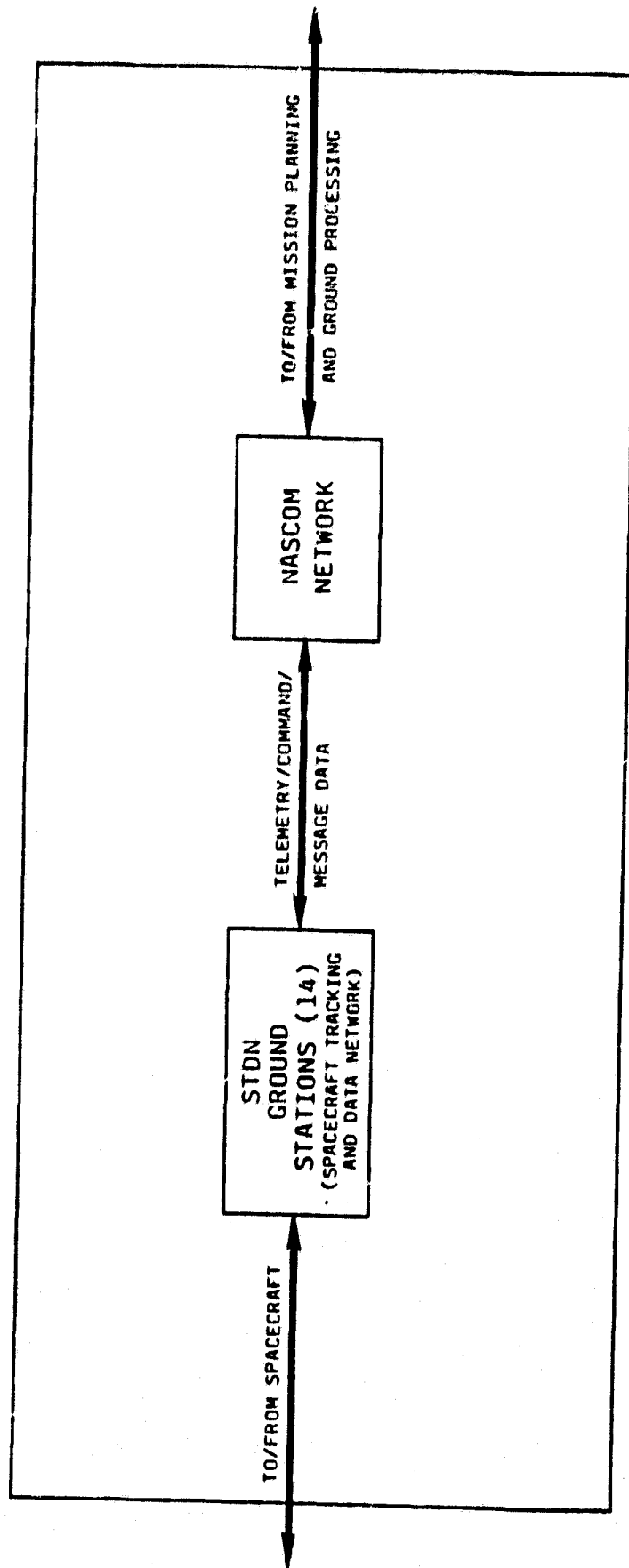


FIG. 2 CURRENT NASA COMMUNICATIONS AND DATA ACQUISITION NETWORK

Inputs, Outputs, and Controls

The inputs to STDN consist of spacecraft commands which are transmitted from the POCC's to the spacecraft. STDN receives schedules from the Mission Planning and Scheduling element of the system.

STDN has several outputs. Tracking data is provided for the Orbit Support Computation Facility (OSCF). Control information and non-image telemetry data are sent to the Network Operations Control Center (NOCC) for monitoring. Telemetry data are received from the spacecraft and transmitted to the POCC's and the SDPF or forwarded during low volume hours over the Domestic Satellite System (DOMSAT). Image telemetry data are stored on tapes and forwarded to the SDPF. STDN produces test data to ensure that proper communications links have been established.

Interrelationships and Constraints

STDN is the interface between the POCC and the spacecraft. In addition, it serves as the interface between the spacecraft and the SDPF. Tracking data are provided to the OSCF.

NASCOM serves as the main communications interface between the various elements of the NASA Space Data System.

C. Mission Planning and Scheduling

Mission Planning and Scheduling, a key control element of the NASA Space Data System, performs the following functions:

1. Plans experiment and spacecraft operations during both the prelaunch and postlaunch (operational) phases of spacecraft missions

2. Schedules:

- a. ground station to spacecraft contacts for receipt and transmission of telemetry and command data
- b. allocation of NASCOM resources
- c. production of user products and their distribution
- d. computation of predicted spacecraft orbit and attitude
- e. command loads to spacecraft that can affect sensor states or change spacecraft attitude or orbit
- f. what sensor data to acquire

3. Controls all the elements of the system

4. Monitors its own performance

5. Accommodates the following:

- a. End User requests for telemetry data (possibly through NASCOM)
- b. End User requests to upload spacecraft commands
- c. POCC resource use requests
- d. requests for special products.

These requests go through a filtering process to limit the degree of resource utilization demands made on the NASA Space Data System.

Structure

The Mission Planning and Scheduling element of the current NASA Space Data System contains the projects that develop, direct, and control spacecraft missions and other elements that direct the planning and scheduling of NASA resources to support and control operational satellites. The major subelements of Mission Planning and Scheduling are as follows:

- 1. Project operations planning
- 2. NOCC

3. Scheduling (Computer-Assisted-Interactive-Resource-Scheduling System --CAIRS-- and Machine-Augmented-Manual-Scheduler --MAMS--).

These elements are responsible for the planning and scheduling of the resources used to connect and support the facilities and functions. Local facilities are responsible for planning and scheduling the use of their own internal resources. Some of the facilities that fall into this category are:

1. Orbit Support Computation Facility (OSCF)
2. Attitude Determination Facility (ADF)
3. Sensor Data Processing Facility (SDPF)
4. Command Management Facility (CMF).

Support functions performed by local facilities but used for scheduling by the Mission Planning and Scheduling element include:

1. Orbit and attitude computation and prediction for spacecraft control and sensor pointing control
2. Command management for the production and control of spacecraft command and on-board computer loads and dumps.

Ground processing functions are scheduled to react to the presence of data or at a time when it is convenient to perform the work or in real time, depending on the mission requirements.

Project Operations Planning

The Project is responsible for the design, development, implementation and operation of the spacecraft. A Project is responsible for the design of both the hardware and software associated with the spacecraft. Space-

craft construction and integration and testing are monitored. A Mission Operations Plan (MOP), which defines the objectives of the mission, and a Support Instrumentation Requirements Document (SIRD), which defines the mission support requested from NASA, are developed. The NASA Support Plan (NSP) defines the support NSAS will provide the spacecraft. After launch, the POCC's are responsible for the operation of the spacecraft under the direction of the project. Generic requirements for the spacecraft are defined. A requirement is generic (as opposed to specific) if the specified activity is intended to occur on an approximately regular basis.

Network Operations Control Center

The NOCC schedules, controls, directs and monitors the activity of STDN and the NASCOM network in support of the various missions.

Network resources are scheduled with the aid of two scheduling systems - CAIRS and MAMS. CAIRS is an on-line system used as an aid in producing conflict-free schedules for network operations. The CAIRS-MAMS systems maintain and update the schedule after it is generated, and they also produce network usage reports. The final schedules are transmitted to all concerned parties over NASCOM at teletype speeds.

Inputs, Outputs, and Controls

The inputs to Project Operations Planning may be divided into prelaunch and postlaunch inputs. During the prelaunch phase of operation, advance information concerning mission goals, science objectives, spacecraft and instrument design characteristics and preliminary mission operations concepts are provided by each Project.

After launch, the Project begins the spacecraft and instrument check-out phase. Daily operations plans are continuously modified depending on the progress made in bringing the mission to an operational status.

Other inputs to Project Operations Planning include spacecraft predicted position, spacecraft power budget, geometric constraints, experimenter requests and commands, and new mission requirements.

The CAIRS and MAMS scheduling systems receive requests for the scheduling of network resources. These requests, spacecraft requirements and predicted positions, ground station capabilities and NASCOM capabilities are necessary for proper scheduling. Each system is used to produce conflict-free schedules for the use of STDN and NASCOM resources.

Interrelationships and Constraints

Project Operations Planning is the focal point of mission planning and development. End Users must make experimenter and command requests to Project Operations Planning. These requests are passed on to the POCC's for execution.

The NOCC must receive requests for NASCOM and STDN resources from the POCC's. The NOCC then utilizes the CAIRS and MAMS systems to produce conflict-free schedules for the network resources.

D. Ground Processing

The receipt, transmission, and processing of telemetry tracking and/or command data are functions of the Ground Processing Element. This element is also responsible for archiving data and distributing it along with any

special user products. It also performs flight maneuver calculations and produces command loads.

Structure

The Ground Processing element of the NASA Space Data System performs ground support functions such as orbit and attitude determination, telemetry processing and archiving, image processing, spacecraft command generation and uplinking, and spacecraft health and safety monitoring. The five functional components which perform ground support tasks are:

1. Sensor Data Processing Facility (SDPF)
2. Attitude Determination Facility (ADF)
3. Orbit Support Computation Facility (OSCF)
4. Command Management Facility (CMF)
5. Project Operations Control Center (POCC).

Sensor Data Processing Facility

The SDPF performs the two major functions of data capture and data formatting. The SDPF is currently archiving data for up to 14 spacecraft with a total of 29 data types and can produce telemetry and image data tapes.

The data capture function of the SDPF is performed by the Telemetry On-line Processing System (TELOPS).

The basic functions of TELOPS are to capture incoming data, pre-edit the received data, and archive edited data.

A Management Information System (MIS) is used to track data through the SDPF. The MIS tracks messages received, files ready for editing, and files ready for archiving.

The Image Processing Facility (IPF) produces image products for the LANDSAT, NUMBUS and Heat Capacity Mapping missions. IPF processing is done in three steps: preprocessing, image processing and product preparation. IPF is managed and controlled by an Information and Production Control System (IPCS).

Attitude Determination Facility

The ADF uses observed spacecraft attitude data obtained from spacecraft telemetry and orbit data obtained from the OSCF to calculate the attitude of a spacecraft.

The five major functions of the ADF are:

1. Real-time attitude determination
2. Attitude control
3. Definitive attitude determination
4. Attitude prediction
5. Bias determination.

Orbit Support Computing Facility

The OSCF uses tracking data acquired by the STDN to provide actual and predicted orbit data for spacecraft. The number of orbit determinations necessary to support each spacecraft depends upon specific spacecraft mission requirements and the occurrence of special or emergency situations:

The six major functions of OSCF are:

1. Input data processing
2. Trajectory/orbit determination
3. Tracking system performance assessment

4. Acquisition data production and validation
5. Scheduling and planning data production
6. Mission maneuver support.

Command Management Facility

The CMF processes requests for spacecraft activity to produce spacecraft commands. The two components of CMF processing are preliminary processing and command loading. Spacecraft activity requests are usually generated by experimenters using the spacecraft's command language. Preliminary processing consists of command editing, merging and assembling.

Project Operation Control Centers

The primary mission of a POCC is to monitor and control the operations and health and safety of a spacecraft on a day-to-day, orbit-by-orbit basis. POCC's may also monitor the health and safety of scientific instruments aboard some spacecraft. The primary services which a POCC provides are:

1. To command a spacecraft
2. To provide information about a spacecraft
3. To act as an interface between the spacecraft and the outside world.

Inputs, Outputs, and Control

The SDPF receives data processing and routing schedules, telemetry data, and image telemetry tapes for processing.

The ADF uses raw telemetry and orbit determination data to produce definitive attitude solutions.

The OSCF uses raw tracking data from STDN and previously generated ephemeris data to produce ephemeris data which are used to produce

planning and scheduling data in various forms.

The CMF receives spacecraft control requests from the POCC's and from End Users. All End User-generated spacecraft control requests must be approved by the appropriate Project Office. The CMF generates spacecraft command loads and ancillary command status information.

Interrelationships and Constraints

The SDPF receives telemetry data in real-time from NASCOM. The ADF and OSCF are used to obtain attitude and orbit data. Telemetry playback can be provided to the POCC's. Finally, telemetry data tapes and image products are provided to the End Users.

The ADF receives generic schedules from the Mission Planning and Scheduling element. Orbit data are obtained from OSCF. Attitude solutions are provided to POCC's and SDPF upon request.

The OSCF receives requests and schedules from Mission Planning and Scheduling. Real-time telemetry and tracking data are obtained from NASCOM. Orbit data are provided to SDPF, NOCC, STDN, and the POCC's.

The CMF receives requests and schedules from mission planning and scheduling. Command data are received from End Users and the POCC's. The CMF sends command loads back to the POCC's.

A POCC interacts with NOCC and other POCC's to negotiate scheduling for spacecraft contact. NASCOM is used to provide real-time contact to the spacecraft. Orbit and attitude data are obtained from the OSCF and ADF, respectively. The POCC's use the CMF to produce command loads. Finally, a POCC may receive telemetry (real-time or playback) then from SDPF or spacecraft.

E. End Users

The End Users are responsible for the functions of establishing mission requirements, performing data analysis, and evaluating sensor operation. In effect, they establish the ground rules under which the mission is conducted and create the criteria for judging mission success. The requirements are input to the system, in both generic and specific terms. In order to perform data analysis, ancillary data such as universal time, satellite position, and sensor evaluation operations are carried out in facilities provided by the users.

The End Users can be divided into two groups; first, the members of the Project staff who are responsible for developing the spacecraft, planning the mission, and carrying out the overall mission operations, and second, the scientific investigators who set the mission goals and objectives, develop the scientific instruments, request modifications to ongoing operations plans and ultimately receive the data collected by their instruments along with the necessary ancillary data so that they can interpret and analyze the results of the experiment.

III. The Future NASA Space Data System

This chapter discusses the future NASA Space Data System for the late 1980's to early 1990's time frame. The general overall conceptual structure of the future system will be very similar to the current system. This chapter's major focus is on the differences between the current and expected future NASA Space Data System.

Overall Conceptual View

The major change that will occur is the source of the input data to the system. For the future, the Tracking and Data Relay Satellite System (TDRSS) is expected to provide the bulk of the telemetry and image data and to result in phasing out all but three of the former ground stations; these three stations operated by and for the Deep Space Network will remain for special dedicated satellite contact support and emergency situations. The TDRSS will provide an increased data transfer capacity and longer spacecraft contact time. However, a significant amount of the data transfer capacity will be used by shuttle vehicles. Multiple concurrent shuttle missions will impose a heavy burden on the data transfer capacity of the TDRSS.

Several of the NASA Space Data System element components are expected to be modified or expanded. The Network Operations Control Center (NOCC) and Mission Planning Center (MPC) functions of the Mission Planning and Scheduling element will be merged into a new function called the Network Control Center (NCC). This center will incorporate the functions of the other two components and will also be responsible for scheduling TDRSS usage. Physical data links will be added to form a network of POC's (POCCNET). The Sensor Data Processing Facility (SDPF) will be upgraded and a new packet switching network (PACOR) capability will be added. The Computer Assisted Interactive

Resource Scheduler (CAIRS) and Machine Augmented Manual Scheduler (MAMS) systems are expected to be replaced by a more fully automated scheduling system. Several ideas developed from the NASA End-to-End Data System (NEEDS) studies may be incorporated into the future system.

A. Constellation of Spacecraft

The makeup of the future constellation of spacecraft will be heavily influenced by the availability of the TDRSS and the shuttle. The TDRSS will provide the tracking and data relay services for all the low earth orbiting spacecraft. Therefore, all these spacecraft must be equipped with antennas, receivers, and transmitters that will allow them to communicate with the TDRS.

As technology progresses the data collection and storage capacity of spacecraft sensors increase. Coupled with improvement and expansion of communications technology and equipment, this produces an increase in the volume of data to be handled by the NASA Space Data System.

The availability of the shuttle, both as a launching vehicle for free-flying spacecraft and a carrier of attached payloads such as the spacelab will introduce the capability of launching payloads on relatively short notice. Also, the large weight-carrying capacity of the shuttle enables very large and complex spacecraft to fly. This will, in turn, lead to the development of much more sophisticated instruments than those now in orbit.

B. Communications and Data Acquisition Network

The Communications and Data Acquisition Network responsibility for the transfer of data between spacecraft and ground facilities will not change. The method of this transfer, however, will change. The NASA

Space Data System will receive an increasing number of requests for real-time data.

Structure

The basic structure of the network will be greatly modified for the future NASA Space Data System. The TDRSS will be installed and serve as the major telemetry and command data transfer mechanism between spacecraft and the ground facilities. As a result of this, 11 of the 14 Spacecraft Tracking and Data Network (STDN) ground stations will be phased out. The remaining three ground stations and the dedicated TDRSS ground station at White Sands will form the new STDN.

NASCOM network hardware will be upgraded. Higher data rate lines will be added to handle the increased bandwidth of TDRSS data transfer as well as the higher data rates of future spacecraft such as LANDSAT-D and Space Telescope. PACOR will be implemented to provide an additional data transfer system. A Shuttle Payload Interface Facility (SPIF) will allow preprocessed shuttle payload data transfer between NASA at Houston and GSFC and may be implemented in this time frame.

Inputs, Outputs, and Control

The basic inputs, outputs, and constraints related to the Communications and Data Acquisition Network will change to reflect new capabilities and facilities. Preprocessed shuttle payload data from Johnson Space Center (JSC) via SPIF will be a new input. The TDRSS will alter the source of input and the destination for output from the STDN system making the total system much more compact. A control and monitor interface link will allow real-time control of data transfer.

Interrelationships and Constraints

The basic interrelationships remain the same as in the future system. NASCOM will work in a state that concentrates data flow between the POCC's, other facilities, and the White Sands ground station that is the dedicated facility for TDRSS.

C. Mission Planning and Scheduling (MPS)

The future Mission Planning and Scheduling element will have the same basic functions as that of the current system. The introduction of the TDRSS represents a new resource that must be managed by the Mission Planning and Scheduling element.

Structure

The Mission Planning and Scheduling element of the future NASA Space Data system is a key control element. The major functional components are as follows:

1. Project Operations Planning
2. NCC Scheduling System (NCCDS)

Internal scheduling of ground system facilities will occur in the same manner as the current system but will be affected by NCC-NCCDS scheduling.

Project Operations Planning

The Project Operations Planning will remain basically unchanged. New projects will be instituted to support new missions.

Network Control Center

The Network Control Center will replace the Network Operations Control (NOCC) and the MPC of the current NASA Space Data System. NCC

will be a real-time network monitoring and control system which is designed to meet the needs of an operational TDRSS. The major capabilities of this system will be as follows:

1. Automated scheduling of network resources
2. Ability to provide a POCC with a total real-time interface with the spacecraft
3. A mechanism for system performance monitoring and evaluation
4. Equipment monitoring and test facilities.

Scheduling System

A new scheduling system within the NCCDS will replace the current CAIRS-MAMS scheduling system. The new scheduler will be automated and will include some conflict resolution procedures. The major components of the new system will be:

1. Scheduler for generic requests
2. Scheduler for specific requests
3. Conflict resolution procedures
4. Resource allocator for both the specific and generic schedule requests
5. An electronic schedule input system which is used to receive scheduling requests from users
6. Automatic and electronic transmission of schedules or responses to scheduling requests to users via NASCOM lines.

The NCCDS is also a complex control and reporting system for NCC functions.

Inputs, Outputs, and Controls

The inputs, outputs, and controls for the future Mission Planning and Scheduling element will be similar to those of the current NASA

Space Data System. The inputs and outputs contain the same type of information although the formats and methodology will change.

Interrelationships and Constraints

The interrelationships of the future Mission Planning and Scheduling element components will be essentially the same as for the current system. The major change which must be considered is the scheduling of the TDRSS resource and the reduction in the number and use of ground stations of the STDN. The scheduling of POCC-to-POCC links for the Project Operations Control Center Network (POCCNET) is a new relationship that imposes constraints on system scheduling.

D. Ground Processing

The Ground Processing element of the future NASA Space Data System has the same responsibilities as the current system. The major changes involve the upgrading of facilities to provide additional processing capabilities.

Structure

Figure 3 illustrates the relationships between the following five functional ground support elements of the NASA Space Data System:

1. Sensor Data Processing Facility (upgraded)
2. Flight Dynamics System (expansion of Attitude Determination Facility -- ADF)
3. Orbit Support Computing Facility (upgraded)
4. Command Management System (upgraded)
5. POCCNET (upgraded and connected POCC's).

ORIGINAL PAGE IS
OF POOR QUALITY

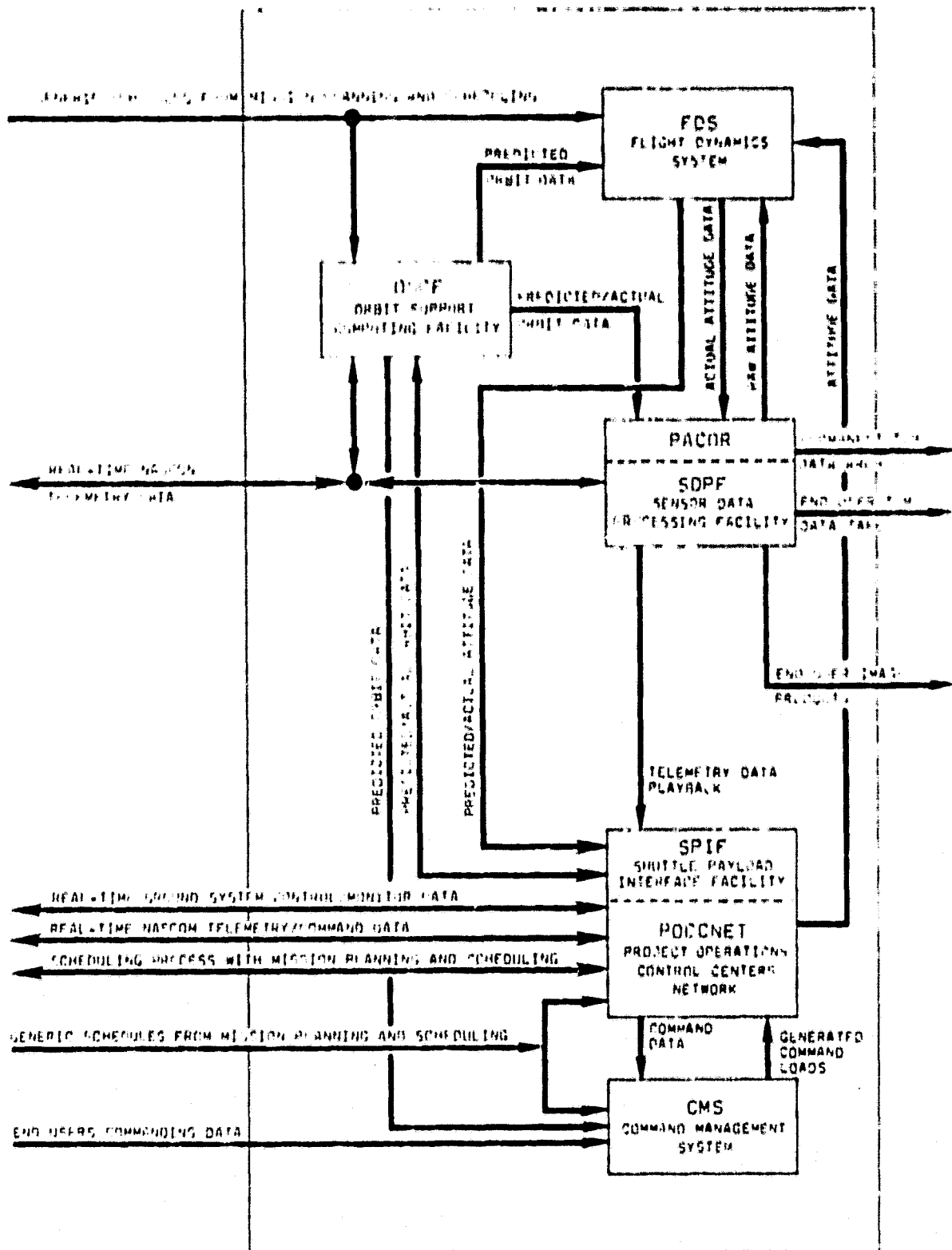


FIG. 3 FUTURE NASA GROUND PROCESSING

Sensor Data Processing Facility

The SDPF will be upgraded through replacement of its computers to provide additional processing capabilities. PACOR will be implemented to allow transfer of information using a packet switching network architecture with its inherent advantages.

The Image Processing (IPF) will continue to exist for currently operational spacecraft. New missions such as LANDSAT-D and Space Telescope will have their own image processing facilities, and will not need the support of IPF.

Flight Dynamics System (FDS)

The Flight Dynamics System is an extension and upgrading of the current system ADF. The new system will provide a more complete support service function for attitude determination and flight maneuvers. Additional capability will be added to assist in TDRSS and spacecraft antenna positioning.

Orbit Support Computing Facility (OSCF)

The computer hardware of the OSCF will be upgraded to provide greater processing capability. Changes are expected during this report time frame to support operations with the TDRSS.

Command Management System (CMS)

The current Command Management Facility will be upgraded through the replacement of its computers to provide additional capabilities. The new CMS facility will perform command processing for an increasing number of spacecraft and provide direct remote input capabilities to the users.

Project Operations Control Center Network (POCCNET)

In the future, POCC capabilities will expand to support multiple simultaneous satellite contacts within a single POCC. Physical data links will be added to form a network of connected POCC's. This expansion and enhancement of capabilities will occur on a piecemeal basis.

Scheduling for the POCC support will require a great deal of coordination between NCC and the POCC's involved to obtain efficient use of the resources. The responsibilities and functions performed by the individual POCC's will remain essentially the same although there is additional internal scheduling for the multiple simultaneous contact capability.

A new ground system control and monitor interface link will be added to relay real-time configuration and data rate change requests. If SPIF is implemented it will be used to interface with the Mission Control Center at JSC to augment payload operations during shuttle support periods.

Inputs, Outputs, and Controls

Almost all of the inputs, outputs and controls remain the same for the future ground processing elements; however, there are minor differences. In the past, image data were transferred from ground stations to the SDPF by magnetic tape. Because of the phasing out of ground stations and the emergence of TDRSS in the near future, image data will be transferred to the SDPF via NASCOM. Electronic transfer of input and output data will be used instead of the current manual methods. Real-time control of data transfer with POCCNET will be possible with the new control and monitor link.

Interrelationships and Constraints

The interrelationships between the ground processing elements are the same as in the current system, with the exception of the Attitude Determination Facility which is being replaced with the Flight Dynamics System. Orbit determination data will be transmitted to the NCC. The scheduling of POCCNET data links institutes a new requirement for the future NASA Space Data System and imposes new constraints on POCC resource availability.

E. End Users

Future system End Users are summarized in Table 1 to the extent that they can be identified at this time. Spacelab and the Space Shuttle will form the bulk of the demand on the future system. It can be anticipated that the system demand will expand beyond that shown here by the time these satellites are actually operational causing a further increase in the data flow load of the future NASA Space Data System.

Table 1. Future Estimate of System End Users

<u>Satellite Name</u>	<u>Number of Experiments</u>
CCE	17
COBE	18
Dynamics Explorer A	37
Dynamics Explorer B	46
ERBS A	7
EUVE	3
Gamma Ray Observatory	56
OSS-1	23
San Marco - D/L	11
San Marco - D/M	2
Space Shuttle (multiple missions)	113
Spacelab 1	142
Spacelab 2	73
ST	62
UARS-2	2

IV. Autonomous Process Control and the NASA Space Data System

This chapter describes the considerations for a recommended new NASA Space Data System. The Dynamic Autonomous System Scheduler (DYASS) concept is introduced to define, defend, and demonstrate a process control structure that automates the now labor-intensive scheduling of resources for the mission support of near-Earth NASA spacecraft for the GSFC. The structure will also provide opportunities to enhance the use of limited resources and make the operations of the system visible and easily controllable from the user and management points of view.

A DYASS System Structure Overview and a logical interconnection of key elements is presented in Figure 4-1. This new structure can improve the information flow of the NASA Space Data System as illustrated in Figure 4-2. To obtain an understanding of how the ultimate system goals relate to this structure, the components, functions, resources and interconnections between each of the logical elements will be discussed. This will form the basis for the introduction of related concepts and technologies that can play a role in achieving the DYASS system goals.

The DYASS Concept of the NASA Space Data System

The Dyass system structure responds to several different inputs and coordinates them to control the use of resources and to produce desired outputs. The system handles the following input categories:

1. User requests
2. Resource allocation
3. Telemetry processing
4. General spacecraft support and control
5. System monitoring

ORIGINAL PAGE IS
OF POOR QUALITY

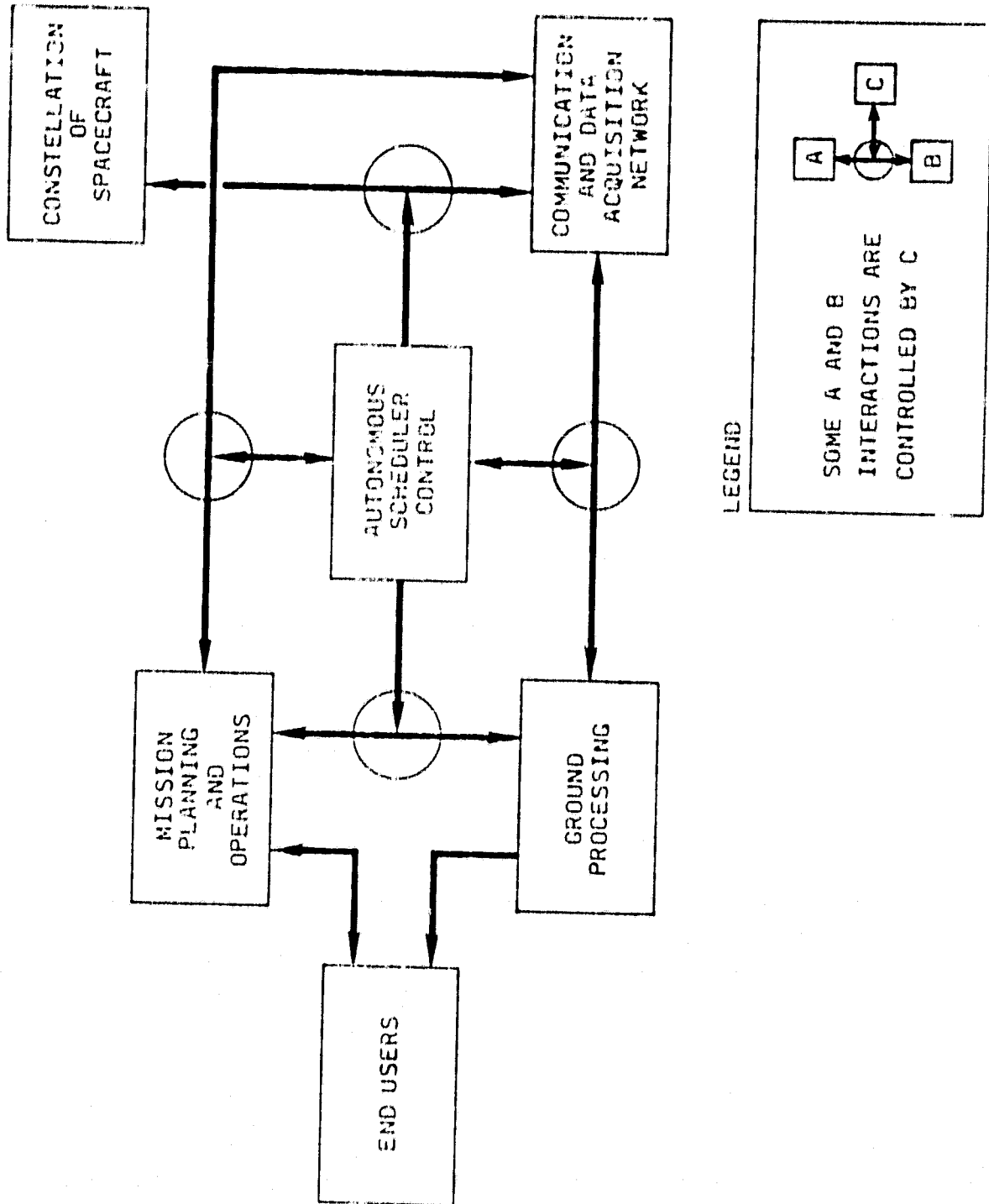


FIG. 4-1 DYASS SYSTEM MAJOR ELEMENT STRUCTURE

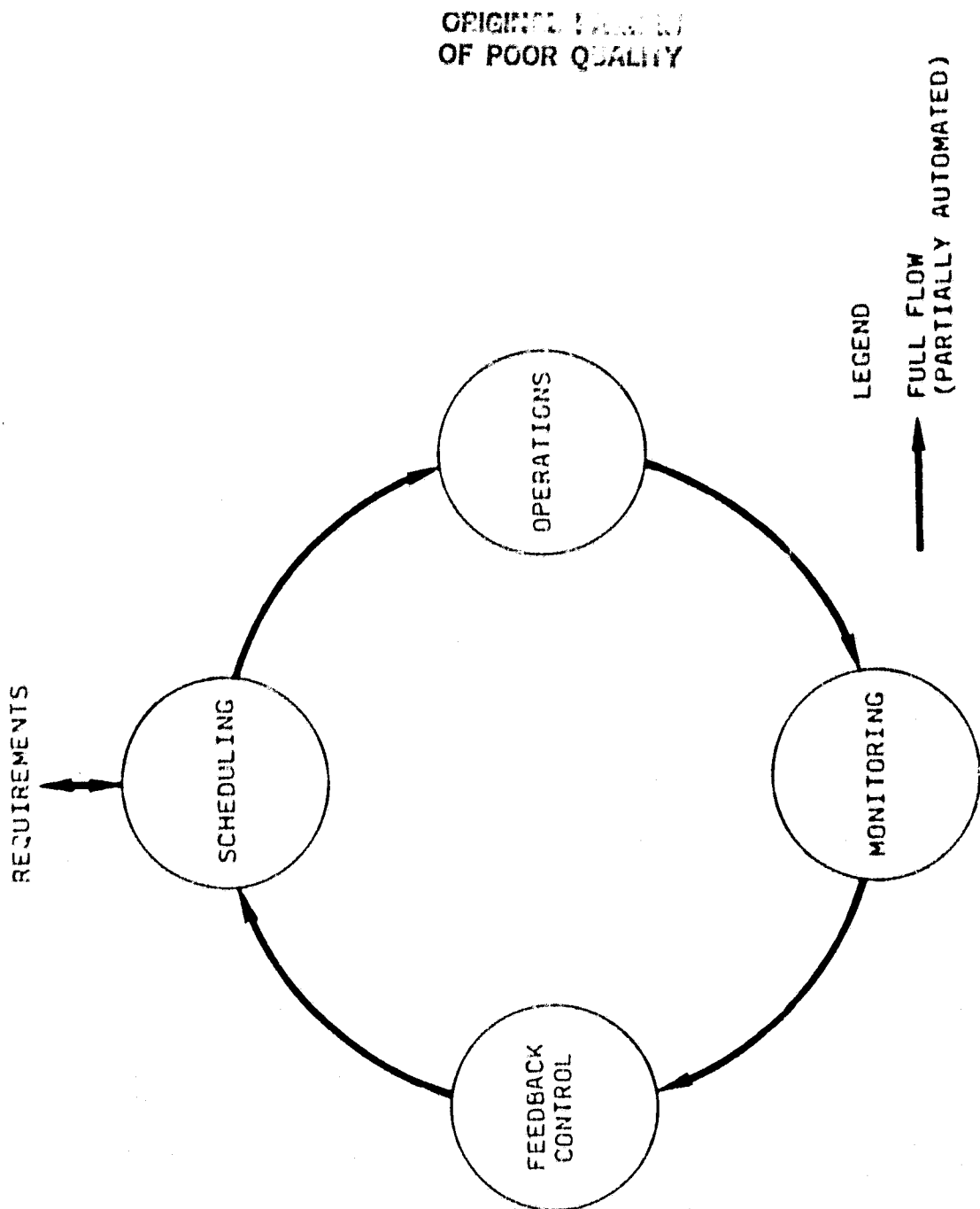


FIG. 4-2 DYASS INFORMATION FLOW

User Requests

The system must be able to deal intelligently with various types of End User requests. These include requests for:

1. Ground system resources (NASCOM, STDN, POCC, SDPF)
2. Experiment control
3. Spacecraft status
4. Spacecraft telemetry for analysis
5. Current status and expected completion time of requests
6. Special products such as imagery and telemetry archive or image data tapes.

Resource Allocation and Scheduling

The system must be able to produce schedules which allocate system resources efficiently. The elements within this category are:

1. DYASS system schedule production and distribution
2. NASCOM allocation
3. Spacecraft contact
4. STDN/TDRSS allocation
5. Project Operations Control Center Network (POCCNET) allocation
6. Attitude
7. Orbit.

Telemetry Processing

The spacecraft sensor data must be processed. The tasks involved are:

1. Telemetry processing
2. Imagery processing
3. Data archiving and distribution.

Telemetry data are processed to determine spacecraft status and archived for distribution.

General Spacecraft Support and Control

DYASS must be capable of controlling and scheduling the general spacecraft support required by the system. These functions are:

1. Orbit computation
2. Attitude computation
3. Spacecraft health and safety monitoring
4. Command and onboard computer management.

System Monitoring and Performance Evaluation

The DYASS system must be capable of monitoring the performance and status of each of its components. Performance evaluation is an internal function of the system and is necessary to check and dynamically refine the effectiveness of the scheduling algorithms. Results of the evaluation can be used to determine areas of the system that may need improvement and to modify tentative schedules.

DYASS System "Job" Concept

A key idea developed to study the DYASS system concept is that of a "job". One of the main functions performed by the DYASS system is the scheduling of resources based on user requests. The user group includes experiment principal investigators, the POCC's, and other NASA agencies. We can equate a single request input to the DYASS system with a single "job" for a computer operating system. The request may cause several actions to occur, usually due to implied or preparatory operations necessary for the completion of the request. We can view these actions as related subtasks of the original request. The structure and ordering of the operations necessary to fulfill the request produces a task graph of connected subtasks whose scheduling must be coordinated to ensure proper subtask sequencing. A task graph is a structure that relates the subtasks in terms of order of

execution.

The processes of schedule production, maintenance, and monitoring can lead to very complex problems. For example, if a key subtask for a request misses its scheduled activation and does not finish on time, it may be necessary to reschedule the remaining lower task graph levels, that is, all of the dependent and pending subtasks. A delay of one or more of a set of parallel scheduled activities can produce the same problem.

Autonomous Process Control

An Autonomous Process Control (APC) system exhibits the following properties.

1. It controls a set of events.
2. The interrelationships between the events can be expressed in the form of a schedule for events that are initiated by time and a task graph for events that are initiated by completion of other events; that is, events are time driven, or event driven and the sequence of events is completely specified.
3. The actions of the system occur automatically without human intervention as specified by the relationships between time and events.

A fully autonomous process control system is not practically realizable, as human intervention is inevitably required. However, it can be approximately realized. The benefits of an APC system include:

1. Minimal dependence on human contact for control or decisionmaking
2. A closed, secure, and thus more reliable system.

The DYASS concept is an attempt to restructure the NASA Space Data System along the lines of an autonomous process control system. Human input and control will always be required for critical decisionmaking. However, as the decision process becomes better understood and decisionmaking logic is added, a true APC system will be asymptotically approached.

V. Scheduling and DYASS Problems

A. Scheduling

Scheduling is a key part of the Dynamic Autonomous System Scheduler (DYASS) concept. The topics discussed include:

1. The scheduling system environment and implications
2. Classes of scheduling problems
3. Technologies and techniques applicable to scheduling problems
4. Scheduling algorithms,

The Scheduling System Environment

A system that uses a scheduling function as a major control element operates in an environment with certain characteristics. The properties can be partitioned into several categories such as:

1. System control
2. Inputs and outputs
3. System scheduling goals
4. Condition handling and error recovery.

Early in the design of the system a decision has to be made about the degree of human intervention and control. Critical decisions require human approval while less important decisions can be made automatically through the use of predefined algorithms. The closer an implementation is to an autonomous process control system the greater the need for communications between the elements.

The inputs to the system consist of event or activity schedule requests ("jobs"), system resource information, resource requirements information, component status information, schedule completion information, and human decisions and control directives. Outputs of the system

include schedule information, rejected requests, current system component status, resource utilization, and request completion information.

The goals of system scheduling are crucial since they determine the types of schedules that are ultimately produced. The most frequently used scheduling goals include:

1. Preservation of the health and safety of system components
2. Maximization of user scheduling request satisfaction
3. Maximization of the use of system resources or a subset of system resources
4. Maximization of system throughput
5. Even distribution of system loading
6. General scheduling rules
 - a. Processing time priority
 - b. Due date priority
 - c. Number of operations priority
 - d. Cost priority
 - e. Setup time priority
 - f. Arrival time priority
 - g. Machine priority
 - h. Weighted priority
 - i. Heuristic methods.

It should be noted that the scheduling goals above are independent and may contend with one another. For example it is well known that maximizing the use of system resources does not necessarily guarantee (and might even hinder) the maximization of system throughput. Setting minimum levels of satisfaction for these goals serves as a basis for selecting an optimal schedule.

The system should be able to handle the following exceptional conditions and errors:

1. Improper job specification
2. Job scheduling deficiencies
3. Resource health and safety emergencies
4. Equipment failure
5. Communications failure
6. Software failure
7. Delays in communications or coordination of operations
8. Shortage of resources.

Some of these conditions cause tasks to miss their scheduled completion time. The delays could cascade down the schedule if corrective action is not taken as soon as possible. There are two types of improper job specification - improper parameter specification and incomplete parameter specification. The three classes of scheduling deficiencies are underestimating and overestimating the time required for a task and system overload. Monitoring and feedback control can help reduce the impact of these problems. The corrective action for an insufficient task time error depends on the system load and the individual request.

If a resource health and safety emergency arises, the system must be able to initiate emergency action and notify the appropriate personnel of the nature of the emergency. The schedule must be altered to meet the needs of the emergency.

Classes of Scheduling Problems

In developing an automated scheduling system concept, it is convenient to compare the DYASS system to classic scheduling problems. Many of the

scheduling problems can be (at least partially) solved through the use of deterministic algorithms. The algorithms are based on the premise that all of the information governing the scheduling decisions are known in advance.

Traveling Salesman Problem

In this classic combinatorial problem, a salesman must devise a route for visiting each of n cities once and only once, returning to the starting city. The traveling salesman problem is ideally suited to such scheduling activities as optimizing the productivity of assembly-lines or optimizing the load/mile value of shipping times. The solution to the problem is a schedule. A variety of algorithms can be used to generate solutions, ranging from linear programming methods to state-space search techniques.

In this generic form, the traveling salesman problem is applicable to only a small subset of the DYASS problem, that of scheduling parallel tasks on a single resource. However, the problem is not so easily extensible to the general DYASS situation of scheduling multiple tasks on a variety of dissimilar resources. Nor is it particularly adaptable to random activity requests.

Elevator Problem

The generic elevator problem involves one or more elevators of finite capacity, a finite number of arrival and destination ports, and a random distribution of service requests. In contrast to the traveling salesman problem, the solution to the elevator problem is not a schedule but rather an algorithm by which a dynamically changing schedule can be generated.

In terms of the DYASS problem, the dynamic aspects are obviously

similar. Moreover, the attributes of finite capacity servers (elevators) and fixed number of ports (spacecraft, POCC's, etc.) are analogous. However, the DYASS attributes of priorities, deadlines, and feasibility windows are not covered, nor is the concept of sequences of activities.

Bin-Packing Problem

The general bin-packing problem concerns a finite set of bins of possible differing capacity, a set of objects to be stored in the bins, and rules for associating objects with one another. In terms of the DYASS problem, a bin can be characterized as a resource/time-interval pair and the objects characterized as a resource/time-interval/time-requirement 3-tuple representing a service request.

It has been shown that the general bin-packing problem is NP-Complete; that is, the number of operations required to compute solutions is strongly believed to grow exponentially with the number of objects. Deterministic methods cannot guarantee the optimal solution without generating all solutions and comparing them. Usually, this is not computationally feasible due to the NP-Completeness of deterministic algorithms, nor is it generally acceptable to suffer the sub-optimal performance of a random solution.

Distribution Problems

The class of problems commonly called distribution problems share rather general attributes of sources and sinks, distribution rates and requirements at the sources and sinks, and (optionally) defined paths for distribution flow. In the DYASS scheduling environment, distribution theory is applicable in maximizing the throughput of the Tracking and Data Relay Satellite System (TDRSS) communications.

Technologies and Techniques Applicable to Scheduling Problems

Many mathematics and computer science concepts can be used to simplify the solution of complex scheduling. Contributions from the areas of Operations Research, Artificial Intelligence, and Data Base technology are particularly important and are discussed in the succeeding paragraphs.

Operations Research

Some of the operations research techniques which are applicable to scheduling are:

1. Dynamic programming
2. Queueing theory.

Dynamic programming is an operations research technique which is used for making a sequence of interrelated decisions. A systemic procedure is provided for determining the combination of decisions which maximize overall effectiveness. A dynamic programming problem includes the following characteristics:

1. The problem can be divided into stages with a policy decision required at each stage.
2. Each stage has a number of states associated with it.
3. The policy decision transforms the current state into a state associated with the next stage.
4. Optimal policy decision may be different at each stage of the problem.

Queueing theory is the branch of operations research which is concerned with the study of waiting lines (queues) to determine effective servicing strategies. The results are dependent upon the particular queueing assumptions that are made. Some of these assumptions are:

1. The queueing discipline, i.e., the criteria for selecting the

next item to be serviced

2. The arrival time of items into the queue
3. The number of processors servicing the queue
4. The service time required by an item.

Queueing Problems and Their Potential Relationship with the DYASS Concept

A wealth of analytic theory applies to queueing systems. Most operating system schedulers are based on results of queueing theory applied to expected and observed environments. Within operating systems, queues are maintained of requests for services from operating system scheduled elements (e.g., memory allocation, I/O, CPU time, etc.). Within the DYASS scheduling environment, queues represent multiple requests for service from DYASS scheduled elements. The similarities between operating system scheduling and DYASS scheduling are immediately apparent. The differences are in the operating environment.

Queueing theory is also applicable to the more subtle problems of system performance analysis and self-improvement. In this context, one use of queueing analysis would be to generate a schedule by whatever means, simulate the activities scheduled, analyze the queues induced in the simulation, and refine the schedule based on the results of the analysis. Another application would be to analyze a schedule and predict the performance.

Data Bases

Data base technology can be applied to facilitate information management in support of a scheduling system. A Data Base Management System (DBMS) can be useful in support of the following functions:

1. Schedule storage and maintenance
2. Request definition and management
3. Performance monitoring
4. Storage and maintenance of scheduling conflict resolution strategies.

DBMS's are currently based on a relational, network or hierarchical model. Relational data bases store information in a table-like format. Queries used to retrieve information may be thought of as algebraic operations. Network data bases use a network of ring structures to store information. Information storage and retrieval is based on key information which is used to locate the correct ring position in the network. Hierarchical data bases are tree structures. Keys are used to retrieve and store information from and to the appropriate sub-trees.

Artificial Intelligence

Artificial intelligence offers several problem solving methodologies which are appropriate to scheduling problems. These are:

1. Heuristic search
2. Action synthesis
3. Backward reasoning
4. Analysis by a set of production rules.

Heuristic search techniques are selective, non-exhaustive techniques. Potential solutions are generated on the basis of their plausibility, based on the knowledge of the problem domain and knowledge ascertained from previous solution attempts.

Action synthesis is the construction of a sequence of component operations to achieve a goal. One would have a set of task domain components

which are combined to obtain a goal.

Backward reasoning or problem reduction partitions a problem into a set of subproblems which are easier to analyze. This process continues until all of the subproblems have been subdivided or solved.

Production rules can be used to define a set of transformation rules to guide the decisionmaking process of a scheduling system. These rules are inflexible in the sense that they do not use ancillary subjective information or past experience in the decisionmaking process. Currently, research is being performed on systems that modify their own production rules based on subjective knowledge, recognized special situations, and collected system performance data.

Artificial intelligence strategies applicable to scheduling include:

1. Interval swapping
2. Resource substitution and sharing
3. Performing a subset of a request.

Interval swapping involves the exchanging of the scheduled time period for two tasks. This is a particularly good strategy when a task's predecessor has not been completed and another task is ready to proceed. This strategy may also be used to reschedule entire requests.

Resource substitution and sharing involves the use of alternative equipment to perform a task, if such equipment is available. This is also an acceptable strategy for conflict resolution.

In certain situations, a request may not be schedulable because a subset of tasks is incompletely specified or cannot be scheduled. An acceptable scheduling strategy is to perform the subset of the request which can be scheduled.

Scheduling Algorithms

For purposes of this discussion, scheduling algorithms are divided into two classes, random requests, and known (fixed) requests. The DYASS environment mixes types of requests but generally treats the scheduling situation as a problem of the second class due to the preponderance of known requests.

Scheduling Random Requests

Several types of algorithms fall into this class. Two types most applicable to DYASS can be characterized as elevator algorithms and operating system algorithms. Among the elevator algorithms there are two major subdivisions that can be characterized as finite-state algorithms (e.g., classic elevator problem) or control algorithms (e.g., classic disk problem). The finite-state problems are amenable to several types of solutions. However, the most popular include automata and state-space representations.

The key to automata solutions is that all outcomes are determined and explicitly represented in the state descriptions of the automata. There is no choice of a next move. That choice is determined by the current state and stimulus. For control problems, the object is to control the state of system variables.

Scheduling Known Requests

Within this category, algorithms can be classified according to whether the resources are scheduled over a continuous interval or discrete partitions (quanta) of time. For either case, an important attribute of the problem is "feasibility." A request for service of a task on a resource is "feasible" if the request is for a time interval during which all activation criteria

are met. For example, a satellite contact is "feasible" between AOS (Acquisition of Signal) and LOS (Loss of Signal) since the actual contact can be scheduled anytime within that interval.

The theory of Interval Arithmetic is applicable for scheduling resource requests on a time continuum. This is especially convenient in evaluating objective functions and in conflict resolution. The missing scheme for conflict resolution can be supplied in a number of forms ranging from human determination to set manipulations on feasibility intervals and requests. Objective functions (other than priority selection) such as uniform distribution of processing are applicable both in conflict resolution and in the primary scheduling algorithm.

Where time intervals have been discretely partitioned for scheduling purposes, the scheduling process is similar to that of memory management with a page allocation scheme. Moreover, similar algorithms are applicable (e.g., First Fit, Best Fit). Since this problem is NP-Complete, there is no feasible way to judge the optimality of the solutions found with any of the algorithms. For example, while it has been shown that in the long run First Fit and Best Fit algorithms allocated equally well, at any one snapshot the Best Fit allocation is no worse and perhaps significantly better than the First Fit allocation. In terms of the DYASS problem, this means that the Best Fit algorithm potentially involves less conflict resolution. The impact of conflict resolution must be weighed against the extra processing required for Best Fit. These two algorithms relate to DYASS in that for resource requests, the span of time quanta (partitions) for which the request is feasible is analogous to the span of memory pages. The conflicts to be resolved correspond to requests on a scheduler wait list.

B. The DYASS Scheduling Problem

The investigation of the DYASS concept as a scheduling system problem is divided into parts. The first part discusses the parameters and constraints inherent in the implementation of DYASS. Then scheduling algorithms and techniques are introduced.

Scheduling Parameters and Constraints

The DYASS concept is a radical departure from the current scheduling and resource control capability. The transition may require a gradual reeducation of current system users. This may be complicated by the phased evolution planned for the implementation of the NASA future system.

Closely related to transition constraints is the composition of the data system environment. At no time is the environment expected to be homogenous. Therefore, the DYASS scheduling, monitoring, and controlling capabilities should be planned around a heterogeneous processing environment, and include separate scheduling activities for the dissimilar system components. Moreover, the separate activities and resultant schedules should be integrated into a master schedule, monitored and controlled as one system-wide schedule.

A state space reduction technique such as the definition of unit schedulable items is particularly desirable in a DYASS schedule implementation. Unfortunately, the Tracking Data and Relay Satellite System (TDRSS) imposes a constraint that prevents the use of the unit schedulable state space reduction technique.

The TDRSS is a potential bottleneck for the NASA Space Data System and a problem that DYASS must resolve. The system will be loaded with shuttle support requirements. Shuttle support requires long periods of

TDRSS contact using large amounts of TDRSS bandwidth capacity. Thus, TDRSS is unavailable for other spacecraft contacts. In addition, the ground facilities of TDRSS may have to store temporarily and forward later large volumes of data from certain high-data-volume spacecraft such as LANDSAT-D and Space Telescope.

The system is further constrained by the impact of the dynamic rescheduling problem. If dynamic rescheduling is to occur, there must be enough time available to ensure that the new schedule can be circulated, and that facilities are properly configured to perform the new schedule. Dynamic rescheduling must not adversely affect any activities which are in progress or which are to begin shortly. There is insufficient time to produce a new optimal schedule. Only minimal impact scheduling can be performed. Response time parameters have to be established for ad hoc requests to define the limits of the scheduling problem. Monitoring and dynamic analysis of the character, frequency, and duration of rescheduling may simplify or reduce its impact.

Applicable Scheduling Algorithms

In the DYASS environment, the scheduling problem is composed of several discrete categories of resources that are linked (in terms of scheduling) by task activity threads that span the set of resources. Within each category of resource (assuming homogeneity within categories), requests are to be scheduled according to some evaluation criteria (objective function). This precludes the use of a single classic scheduling algorithm to solve the global scheduling problem.

Although not generally applicable to the global DYASS scheduling problem, the Traveling Salesman can be applied to some isolated DYASS

subproblems. The algorithm realizes the objective function of minimizing the total time wasted transitioning from one program to another. For this it is assumed the triangle inequality holds.

Two algorithms of the elevator problem class are applicable to DYASS. A great variety of bin-packing algorithms are applicable to the DYASS scheduling problem.

As its name implies, the Earliest Deadline Algorithm schedules processes according to the relative occurrence of process deadlines. Since most scheduled activities in the DYASS environment are subject to deadlines, this algorithm is potentially directly applicable. Drawbacks to employing this concept directly are that while most activities are deadlined, in the TDRSS environment the period of view is so relatively long that some deadlines are insignificant in the scheduling process. Also, while the algorithm supports scheduling on multiple resources it does not immediately extend to the global DYASS environment of scheduling activity threads across resources categories. In one form, the Earliest Deadline Algorithm is similar to the Request Priority Algorithm with the earliest deadline analogous to the highest priority.

In the Least Laxity Algorithm, the concept is to schedule first those requests that most limit resource excess capacity. That is, given an increment of time already allocated and a set of requests still pending, the next request to be scheduled is that which is most restrictive in terms of excess resource capacity (maximizes the schedule laxity for the remaining requests). When all schedule requests are known a priori, as well as the processing requirements, the Least Laxity Algorithm is optimal in the sense

of accommodating requests. However it is not immediately extensible to cover the global DYASS problem of scheduling activities across multiple heterogeneous resources.

Recognizing the TDRSS links as a major bottleneck in the throughput capacity of the system, a prime target for optimization is the scheduled use of TDRSS. For this, the techniques of flow distribution are particularly effective. However, this requires a major assumption in the scheduling process: the scheduled items and source and sink capacities must be held constant for some time interval over which optimization is to be achieved. Optimal intervals may be determined dynamically via any of several heuristic techniques or may be chosen analytically and fixed. The actual flow distribution algorithm to be used would depend on the satellite constellation for the interval being scheduled. Alternatively, dynamic programming techniques could be employed to create the algorithm based on the environmental conditions. To integrate this process into the global scheduling program, one approach would be to maximize the contacts to drive the backward scheduling of all required predecessor activities.

VI. Artificial Intelligence

NASA is, to a significant degree, an agency devoted to the acquisition, processing, and analysis of information - about the Earth, the Solar system, the Stars, and the Universe. The principal goal of NASA's booster and space vehicle commitment is to acquire such scientific information for the benefit of the human species. At the present time, the amount of data made available by NASA missions is larger than scientists can comfortably sift through. A typical information acquisition rate in the 1980's is about 10^{12} bits per day for all NASA systems. We have reached a severe limitation in the traditional way of acquiring and analyzing data.

With machine intelligence and modern computer graphics, an immense amount of data can be analyzed and reduced to present the scientific or technological results directly in a convenient form. With the successful launch of the Space Shuttle, the space program is at the threshold of a new era. This will enable expanded space industrial activities and, by the end of this century, could lead to Satellite Power Systems for solar energy production and to manned space stations for commercial processing and manufacturing in space. A major objective for NASA is to develop the enabling technology and to reduce the costs for operating such large-scale systems during the next two decades. There are many simple or repetitive tasks which existing machine intelligence technology is fully capable of dealing with more reliably and less expensively than if human beings were in the loop.

Machine intelligence and robotics are not only relevant but essential to the entire range of future NASA activities. Content analysis of Earth

orbital and planetary spacecraft results is merely one application. Other applications exist: in mission operations, in spacecraft crisis management, and in large constructions in Earth orbit or on the Moon.

The uses of robotics can be broadly grouped into manipulators and intelligent planetary explorers. There already exist automatic vision and manipulation techniques that could be developed into practical systems for automatic inspection and assembly of components. Intelligent robot explorers will become imperative, if sophisticated large-scale interplanetary exploration is to become a reality.

Software developed within NASA is often done in a batch environment using punched cards, resulting in a turnaround time of hours or even days. In contrast, the machine-intelligence laboratories are characterized by being totally on-line and interactive. The investment made to substitute computer processing for many manual activities of programmers should ultimately result in improved software quality and programmer productivity.

There are several data management issues where artificial intelligence techniques could be brought to bear. These areas range from the control of data acquisition and transmission, data reduction and analysis, and methods for dissemination to users. For example, onboard computers should perform data reduction and selective data transmission. This will minimize the amount of data transmitted and conserve communication channels and bandwidth. This requires an advanced computer capable of various types of data analysis. Once the data reaches a ground collection site, there are three types of data management functions required to make the data accessible and usable to researchers. First, the data must be archived. Secondly, access to specific

portions or collections of the data, locating predetermined criteria must be provided. Both archival and criteria selection management systems are well within current technology. However, the third type of database management function, the ability to access data by its content, does not yet exist, and requires specific artificial intelligence support. It would utilize a Knowledge Base containing specific facts about the data, general rules concerning the relationships between data elements and world models into which complex requests can be evaluated. This Knowledge Base would guide the system in locating data containing the desired attributes utilizing a predefined indexing criteria and the relationship of the desired attributes to the indexing attributes.

AN OVERVIEW OF KNOWLEDGE REPRESENTATION

This is a brief overview of terminology and issues related to Knowledge Representation (KR).

Knowledge Representation is a central problem in Artificial Intelligence (AI) today. Its importance stems from the fact that the current design paradigm for "intelligent" systems stresses the need for expert knowledge in the system along with associated knowledge-handling facilities. This paradigm is in sharp contrast to earlier ones which might be termed "power-oriented" in that they placed an emphasis on general purpose heuristic search techniques.

The basic problem of KR is the development of a sufficiently precise notation for representing knowledge. We shall refer to any such notation as a (knowledge) representation scheme. Using such a scheme one can specify a knowledge base consisting of facts. For the purposes of this paper, a knowledge base will be treated as a model of a world/enterprise/slice of reality.

Representation Schemes

Representation schemes have been classified into declarative and procedural ones. For the purposes of the discussion that follows, we further subdivide declarative schemes into logical and (semantic) network ones.

A. Logical Representation Schemes

Such schemes employ the notions of constant, variable, function, predicate, logical connective and quantifier to represent facts as logical

formulas in some logic (First or Higher Order/Multi-valued/Modal/Fuzzy etc.). A knowledge base, according to this view, is a collection of logical formulas which provides a partial description of a world. Modifications to the knowledge base occur with the introduction/deletion of logical formulas so logical formulas are the atomic units for knowledge base manipulation in such schemes. The use of Logic as a representation scheme can be traced at least as far back as McCarthy's "Advice Taker".

An important advantage of logical representation schemes is the availability of inference rules in terms of which one can define proof procedures. Such procedures can be used for information retrieval, semantic constraint checking and problem solving.

Another strength of logical schemes is the availability of a clean, well-understood and well-accepted formal semantics, at least for "pure" logical schemes that are quite close to First Order Logic. As one moves to representation schemes that try to deal with knowledge acquisition, beliefs and defaults the availability of a clean formal semantics becomes more problematic and is an area of active research.

A third strength of logical schemes is the simplicity of the notation employed which leads to knowledge base descriptions that are understandable. Another advantage is the conceptual economy encouraged by logical representation schemes which allow each fact to be represented once, independently of its different uses during the course of its presence in the knowledge base.

A major drawback of logical schemes is the lack of organizational principles for the facts constituting a knowledge base. A large knowledge base, like a large program, needs organizational principles to be understandable as a unit. Without them, a knowledge base can be as unmanageable

as a program written in a programming language which does not support abstraction facilities.

A second drawback is the difficulty in representing procedural and heuristic knowledge such as

"If you are trying to do A while condition

B holds, try strategies C_1, C_2, \dots, C_n ".

An interesting departure from logical representation schemes has been proposed by Kowalski who argues in favor of a dual semantics for logical formulas of the form

$$B_1 \wedge B_2 \wedge \dots \wedge B_n \supset A$$

The first is the traditional Tarskian semantics. The second is a procedural semantics which interprets the formula as

"If you want to establish A, try to establish

B_1 and B_2 and ... and B_n ".

The language PROLOG realizes this idea and has gained many supporters as it combines advantages from logical and procedural representation schemes.

B. Network Representation Schemes

Such schemes, often called semantic networks, attempt to describe a world in terms of objects (nodes) and binary associations (labelled edges), the former denoting individuals and the latter binary relationships in the world being modelled. According to a network representational view, a knowledge base is a collection of objects and associations, or a directed labelled graph, and modifications to the knowledge base occur through the insertion/deletion of objects and the manipulation of associations. Semantic networks have gained wide acceptance as means of modelling human memory and as useful representations for building "intelligent" systems.

Early versions of network schemes tended to encourage a proliferation of association types (edge labels) as new kinds of knowledge were represented. This practice and other deficiencies of earlier network schemes have been criticized in (Woods 75) and (Schubert 76). Their criticisms have triggered a trend towards network schemes with a fixed number of primitive association types which have well-defined semantics and are descriptively adequate in that they can be used to represent any fact expressible in a logical scheme. Some of these schemes simply view network knowledge bases as convenient implementations of logical ones. Others view network schemes as tackling a different set of representational issues and propose a set of primitive association types accordingly.

Due to their nature, network schemes address directly issues of information retrieval since associations can be used to define access paths for traversing a network knowledge base. Another important feature of network schemes is the potential use of primitive association types such as those mentioned above for the organization of a knowledge base. A third advantage is the obvious graphical representation of network knowledge bases which enhances their understandability.

A major drawback of network schemes has been the lack of a formal semantics and a standard terminology. This is at least partly due to the fact that semantic networks have been used as representational tools in very different ways.

C. Procedural Representation Schemes

Such schemes view a knowledge base as a collection of procedures expressed in some language. Most procedural schemes have been influenced quite heavily by LISP which has been used almost exclusively as the implementation language for "intelligent" systems. Indeed, in the past LISP

itself was a favorite representation scheme due to, among other things, its purely symbolic nature and the dynamic run-time environment it offers its users.

Procedural schemes beyond LISP can be classified on the basis of the stand they take with respect to two issues. The first is concerned with the activation mechanism offered for procedures, while the second involves the control structures offered by any one scheme.

On the first issue, PLANNER introduced the notion of pattern directed procedure invocation. A knowledge base is viewed in PLANNER as a global database of assertions and a collection of theorems (or demons) which watch over it and are activated when ever the database is modified or searched. Each theorem has an associated pattern which, upon the theorem's activation, is matched against the data about to be inserted/removed or retrieved from the database. If the match succeeds, the theorem is executed. Thus with theorems the usual procedure calling mechanism is replaced with one where procedures are called whenever a condition is satisfied.

Production systems offer a procedural scheme that is in many ways similar to PLANNER. A knowledge base is a collection of production rules and a global database. Production rules, like theorems, consist of a pattern and a body involving one or more actions. The database begins in some initial state and rules are tried out in some prespecified order until one is found whose pattern matches the database. The body of that rule is then executed and matching of other rules continues.

There are major differences between the activation mechanism of a PLANNER theorem and a production system rule as well. The order in which theorem patterns are matched is undetermined in PLANNER (although the user can define one for any particular situation where he tries to tamper with

the database). "Standard" production systems, like Markov algorithms, have a fixed ordering of rules which determines when will each rule be matched against the database. Another important difference is that theorems can call directly other theorems while productions can only do so indirectly by placing appropriate information on the database. Thus, a production system database can be viewed as a workspace or a bulletin board which provides the only means of communication between rules.

Turning to control structures, there exist several proposals which extend or otherwise modify the usual hierarchical control structure of LISP or ALGOL. As indicated in the previous paragraph, production systems offer one where there is no direct communication or control between rules. Thus a production system knowledge base consists of a collection of loosely coupled rules and this feature renders such knowledge bases fairly easy to understand and modify.

PLANNER's control structure for theorems uses backtracking in that when a theorem's body is executed and fails to achieve a predetermined goal, the side-effects of the unsuccessful theorem are erased and other theorems are tried until one is found that succeeds. It has been argued quite convincingly that backtracking is an unwieldy control structure.

Procedural schemes have in principle one major advantage and one major disadvantage compared to declarative ones. They allow the specification of direct interactions between facts thus eliminating the need for wasteful searching. On the other hand, a procedural knowledge base, like a program, is difficult to understand and modify.

D. Frame-based Representation Schemes

Since 1975, when Minsky originally proposed it, the notion of frame has

played a key role in KR research. A frame is a complex data structure for representing a stereotypical situation such as being in a certain kind of living room or going to a child's birthday party. The frame has slots for the objects that play a role in the stereotypical situation as well as relations between these objects. Attached to each frame are different kinds of information such as how to use the frame, what to do if something unexpected happens, default values for its slots etc. A knowledge base is now a collection of frames organized in terms of some of the organizational principles discussed earlier but also other "looser" principles such as the notion of similarity between two frames.

The original frame proposal was nothing but a framework for developing representation schemes which combined ideas from semantic networks, procedural schemes, linguistics etc. Several representation schemes proposed since then have adapted the frame proposal. Below we present brief descriptions for four of them.

1. FRL (Goldstein and Roberts 77)

An FRL knowledge base consists of frames whose slots carry information such as comments on the source of a value bound to the slot, a default value, constraints, and procedures that are activated when a value is bound, unbound or needed for a slot. All frames are organized into a hierarchy which appears to be a combination of classification and generalization. The procedures attached to a slot are expressed in LISP.

2. KRL (Bobrow and Winograd 77)

This is a more ambitious project than FRL. Like FRL, the basic units of a KRL knowledge base are frames with slots and several kinds of information attached to each slot. Unlike FRL where this information provides details about how to instantiate a frame, KRL is much more concerned with a matching

operation for frames. All on-going processes at any one time are controlled through a multiprocessor agenda which can be scheduled by the designed of the knowledge base. KRL also supports belief contexts which can serve to define an attention focusing mechanism. "Self knowledge" can be included in a knowledge base by providing description about other descriptions.

3. OWL (Szolovits et al. 77)

Unlike other frame-oriented schemes, OWL bases its features on the syntactic and semantic structure of English, taking as founding principle the Whorfian Hypothesis that a person's language plays a key role in determining his model of the world and thus in structuring his thought. An OWL knowledge base can be viewed as a semantic network whose nodes are expressions representing the meaning of natural language sentences. Each node, called a concept, is defined by a pair (genus, specializer) where "genus" specifies the type or superconcept while "specializer" serves to distinguish this concept from all other concepts with the same genus.

4. KLONE (Brachman 79)

A KLONE knowledge base is a collection of concepts where concept is a highly structured object, having slots to which one can attach a variety of information (defaults, modalities etc.). To a concept one can also attach structural descriptions which express constraints on the values that can be bound to the different slots of the concept. Concepts provide purely descriptive structure and make no assertions about existence of a referent or coreference of descriptions. A separate construct called a nexus is used to make assertions about the world being modelled. Also, KLONE offers procedural attachment as a means of associating procedural information, expressed at this time in LISP, with a concept.

Distinguishing Features of Representation Schemes

Below we list some of the more technical (and less vague) characteristics of representation schemes which appear to distinguish them from their semantic data model/program specification language cousins.

1. Multiple Uses of Facts

Unlike a database, whose facts are used almost exclusively for retrieval purposes or a program whose facts are used in the execution of some procedure, a knowledge base contains facts which may have multiple uses. A representation scheme must take this into account in terms of the tools it offers. Below we list some possible uses.

2. Inference

Given a collection of facts, new facts may be deduced from them according to some fixed rules of inference without interaction with the outside world. Some inferences have the flavor of inference techniques in formal logic. For knowledge basis, however, it is also useful sometimes to derive facts through specialized procedures that use other known facts only in fixed ways. For example, a procedure that determines whether a pair is in the transitive closure of some binary relation can perform inferences of a very specialized nature and is only applicable to facts associated with a transitive relation. Also, a knowledge base may be represented in such a way that there are "preferred inferences". The use of defaults is a good example of such a mechanism.

Deduction, with a formal, special purpose or heuristic flavor, is not the only kind of inference. There can also be inductive inferences and abductive ones which have played a role in some knowledge bases.

Given all this variety for inference mechanisms, the question for the

designer of a representation scheme is not how he can include all of them in his scheme, but which ones, if any, he is going to include. Logical schemes clearly have an advantage over other types of schemes when considered from the point of view of (general purpose) inference facilities.

3. Access

Access (and storage) of information in a knowledge base for question-answering purposes constitutes an all-important use of the knowledge base. The associationist viewpoint of network schemes, particularly their organizational principles, make them strong candidates for access-related uses.

4. Matching

Matching as a knowledge base operation can be used for a variety of purposes, including (i) classification, i.e. determining the type of an unknown input, (ii) confirmation where a possible candidate to fit a description is matched against it for confirmation purposes, (iii) decomposition where a pattern with a substructure is matched against a structured unknown and the unknown is decomposed into subparts corresponding to those of the pattern, (iv) correction where the nature of a pattern match failure leads to error correction of the unknown input.

The matching operation itself can be (i) syntactic where the form of the unknown input is matched against another form, (ii) parametric in the tradition of Pattern Recognition research, (iii) semantic where the function of the components of the pattern is specified and the matcher attempts to find elements of the input to serve this function, (iv) forced as in MERLIN where a structure is viewed as though it were another and matches of corresponding items may be forced.

KRL has paid special attention to matching as a knowledge base operation.

Incompleteness

Except for situations where a knowledge base models artificial "micro-worlds", it cannot be assumed that the knowledge base is a complete description of the world it is intended to model. This observation has important consequences for the operations defined over a knowledge base (inference, access, matching) as well as the design methodologies for knowledge bases.

Until recently much of the work of KR ignored the problem of incompleteness or dealt with it in an ad hoc way. Recent work attempts to correct this situation.

Viewing a knowledge base as an incomplete and approximate model of a world which can always be improved but can never be quite complete, leads to design methodologies for knowledge bases which are drastically different from ones for programs. Thus in Programming Language the leading design methodologies stress "once and for all" designs where the designer sits down with a clear idea of the algorithm he wants to realize and by the time he stands up the design is complete. In AI, a knowledge base is developed over a period of time that can be as long as its lifetime through different knowledge acquisition processes that can range from interactive sessions with an expert to the automatic generation of new facts based on the system's "experiences". Organizational principles underlying the structure of a knowledge base can play a crucial role in determining the direction of knowledge acquisition, i.e. which facts should be acquired first and which ones later.

Self Knowledge

There are many kinds of self knowledge. Facts which describe the form or allowable configurations of other facts (e.g. type definitions) are an important kind of self knowledge. Making such facts available for question answering and inference by representing them the same way as other facts is an important capability of declarative schemes which is generally not shared by procedural ones. A good example of use of such self knowledge for knowledge acquisition is provided in TEIRESIAS.

A second kind of self knowledge involves the ability of a system to answer elementary questions about its actions as in SHRDLU, or about the strategies it uses to perform some task as in HACKER.

CONCLUSIONS

There are signs today that KR is maturing at least to the point where there is some agreement on issues and open questions. One can find several knowledge-based systems which perform at an expert or near expert level. There is even some discussion on issues related to Knowledge Engineering which appears to suggest that design methodologies for knowledge bases are following a similar path as design methodologies for large programs, perhaps with a 8 - 10 year lag.

The "NUDGE" System

NUDGE is a knowledge-based office scheduling program developed at the MIT Artificial Intelligence Laboratory. This program accepts informal scheduling requests and produces a schedule containing conflicts and a set of strategies for conflict resolution. A knowledge data base is used to expand and debug schedules. The data base contains data for:

- A. Supplying missing request details
- B. Inconsistency resolution
- C. Determination of available conflict resolution strategies
- D. Determination of necessary task prerequisites
- E. Planning for expected outcomes.

NUDGE attempts to schedule by defining a four item property list which categorizes the scheduling of an event. The list consists of:

- A. The time of the scheduled activity
- B. The activity being scheduled
- C. The resource being scheduled
- D. The object being scheduled for activity.

Each of these four items consists of a heirarchy of sub-items which aid in the definition of an event being scheduled. The goal of NUDGE is to completely define these four components and the interrelationships among them to form a frame for the scheduling request. NUDGE uses its knowledge base to form a frame for the scheduling request by attempting to complete informal user scheduling requests. At this point one has a schedule which could potentially contain conflicts. The scheduling phase of NUDGE is handled by a program called BARGAIN which uses traditional decision analysis programs that have been augmented by Artificial Intelligence techniques to

control the search processes involved in conflict resolution and scheduling.

Initially, a schedule which contains the entire set of conflicts is produced for BARGAIN. These conflicts are then resolved individually using resource-driven or purpose-driven conflict resolution techniques. The resource-driven strategy attempts to alter the particular interval when an event is scheduled while still maintaining the particular event requirement. Purpose-driven techniques analyze the main goal of a scheduled event and modify, or possibly eliminate, requirements which are lesser goals of a scheduled event. Some of the conflict resolution strategies used include:

- A. Relaxing defaults
- B. Sharing resources
- C. Swapping intervals
- D. Relaxing preferences
- E. Eliminating requirements
- F. Substituting resources
- G. Dividing requests into subsets
- H. Using traditional scheduling algorithms.

The overall NUDGE scheduling strategy is to maximize the number of successfully scheduled requests. The implementation of this system has produced good results, and its performance will improve as better conflict resolution strategies and algorithms are employed.

The "I-SPACE" System

The "I-SPACE" system is a man-machine system designed to interface both technical and non-technical professionals to the large, dynamic and diffuse "information space" in which they conduct their daily professional activities. This shares some goals with Goldstein's "PIE" project and

Newell's "ZOG" project, both advanced man-machine information systems. However the I-SPACE system is more aimed at the real-time acquisition and synthesis of information from diverse, often geographically distributed, sources.

The main goal of an I-Space is to deliver a simple interface through which the user gains access to virtually any information he deems relevant to his job. A second goal is that information be delivered in a dynamic fashion which often translates to real-time.

I-Space frames are LISP data structures which contain all the information required to set up, synthesize, and display information from a set of information conduits. Structurally, a frame is a collection of named slots. Visually, a frame is presented by displaying all its slots at an appropriate level of activation for that user. Each slot is presented as a three-window display: the slot name, the slot value/work area and a status region. Shapes and locations of windows and whether or not all three are displayed at the current level of activation are determined by information in the frame's representation. Three levels of frame/slot activation are possible: "browse", "focus", and "invoke". Invoking a slot is tantamount to computing. Since all activation procedures are un- restricted LISP functions, the effects on the I-Space of an invoked slot can be arbitrary.

Scheduling and Processes

At any given moment, the user will have a currently browsed or focused frame (some of whose slots may have been invoked) and possibly a collection of other focused and partially invoked frames in the background. Each slot

has been activated at some level. If the slot's activation procedure is one which is concerned with repetitive or real-time updating, or is on the lookout for certain events or data in the I-Space at large, then it will be making demands on the I-Space for periodic attention. For this reason, the system must include a scheduler process per user, as well as a system-level scheduler for mediating the user-scheduler's requests of the I-bank. As the I-Space shell interprets user desires, and slots become activated, slot activation procedures run and can send scheduling tasks to the U-Scheduler, a separate UNIX process that manages its one user's scheduling needs. The U-Scheduler in turn makes appropriate demands on the S-Scheduler, a system-wide UNIX process which synthesizes all U-Scheduler requests, prioritizes them, then passes them to ZMOB, which finally carries out the requested interactions with the outside.

Another topic of interest that arises because of the I-Space's blend of a PIE-like environment with real-time scheduling contains the channel to be used, the procedure to be run on that channel, the priority required to run it, the arguments to be passed to the procedure, the schedule the procedure should be run by and the consumers that describe what to do when results are returned from the procedure. If there is no channel, the procedure is understood to be special procedure that is handled by the I-SHELL which is the main operating system in the I-SPACE.

The "I-SPACE" system will be a foundation for an eventually very intelligent distributed information system and will represent an incremental advance in interfacing humans with large computer-based information systems.

The following tasks are proposed:

- #1 To investigate the "NUDGE" system and to find better conflict resolution strategies and algorithms.
- #2 To implement the "NUDGE" or "NUDGE TYPE" system to tackle the DYASS scheduling problems.
- #3 To study in detail the "I-SPACE" system.
- #4 To implement the "I-SPACE" system to solve the DYASS scheduling problems.

BIBLIOGRAPHY

GSFC Documents

1. NASA Data Systems Performance Description - 1979 Configuration, CSC/TR-79-6023 under Contract NAS 5-24300, Task 98500, January 1980.
2. NASA End-to-End Data System (NEEDS) Study Plan, MDOD-1SSP/0177, GSFC, January 1977.
3. Computer Assisted Interactive Resource Scheduler (CAIRS) System Description, NASA, GSFC, July 1974.
4. Spacecraft Tracking and Data Network (STDN) Schedule Requirements for the 1980's Based Upon User Mission Needs - Final Study Report, R. Burns, Contract NAS5-23437, Mod. 19 TR 1311, March 1978.
5. NASA Data Systems Performance Description - 1982-84 Configuration, CSC/TM-80-6098, Contract No. NAS 5-24300, Task 98500.

Papers

1. NUDGE, A Knowledge Based Scheduling Program, I. Goldstein and R. Roberts, Proceedings, Fifth International Joint Conference on Artificial Intelligence, MIT, Cambridge, Mass.
2. Bargaining between Goals, I. P. Goldstein, Proceedings of the Fourth International Joint Conference on Artificial Intelligence, 1975.
3. Toward the Realization of Intelligent Controls, G. N. Sardis, Proceedings of the IEEE, Vol. LXVII, Number 8, August 1979.

Continuation of Bibliography

Texts

1. Winston, P. H. and Brown, R. H., Artificial Intelligence: An MIT Perspective. Vol. I and II. MIT Press, 1979.
2. Nilsson, N. J., Principles of Artificial Intelligence. Tioga Publishing Co., 1980.